Student: Victor-Petru Almasan

**Group: 30235**

Table of Contents

1. Requirements Analysis 3

1.1 Assignment Specification 3

1.2 Functional Requirements 3

1.3 Non-functional Requirements 3

2. Use-Case Model 3

3. System Architectural Design 4

4. UML Sequence Diagrams 6

5. Class Design 7

6. Data Model 7

7. System Testing 8

8. Bibliography 9

1. Requirements Analysis

# Assignment Specification

Aceasta aplicatie este destinata unei farmacii. Aplicatia ajuta farmacia in gestionarea facila a angajatilor, medicamentelor si vanzarilor.

# Functional Requirements

Aplicatia are doua tipuri de useri: *administrator* si *user normal*.

*Administratorul* poate sa vizualizeze/editeze/adauge/stearga angajati. De asemenea el poate sa vizualizeze/editeze/adauge/stearga medicamente. Administratorul are si posibilitatea de a genera rapoarte in doua formate: pdf si cvs.

*Userul normal* poate sa caute medicamentele din farmacia dupa nume, ingredient si producator si poate sa vanda aceste produse.

# Non-functional Requirements

* *Update automat la lista userilor in interfata*
  + Source of stimulus: Administrator
  + Stimulus: Interfata trebuie sa fie prietenoasa la adaugarea unui user
  + Environment: Logare ca administrator
  + Artifact: AdminLayoutController
  + Response: Lista tuturor userilor este actualizata
* *Calcularea totalului unei vanzari*
  + Source of stimulus: User
  + Stimulus: Interfata ofera vizual ce fel de constrangeri trebuie aplicate
  + Environment: Logare ca user normal
  + Artifact: UserLayoutController
  + Response: O fereastra cu confirmarea vanzarii medicamentului

2. Use-Case Model

Use case: Operatii pe farmacisti

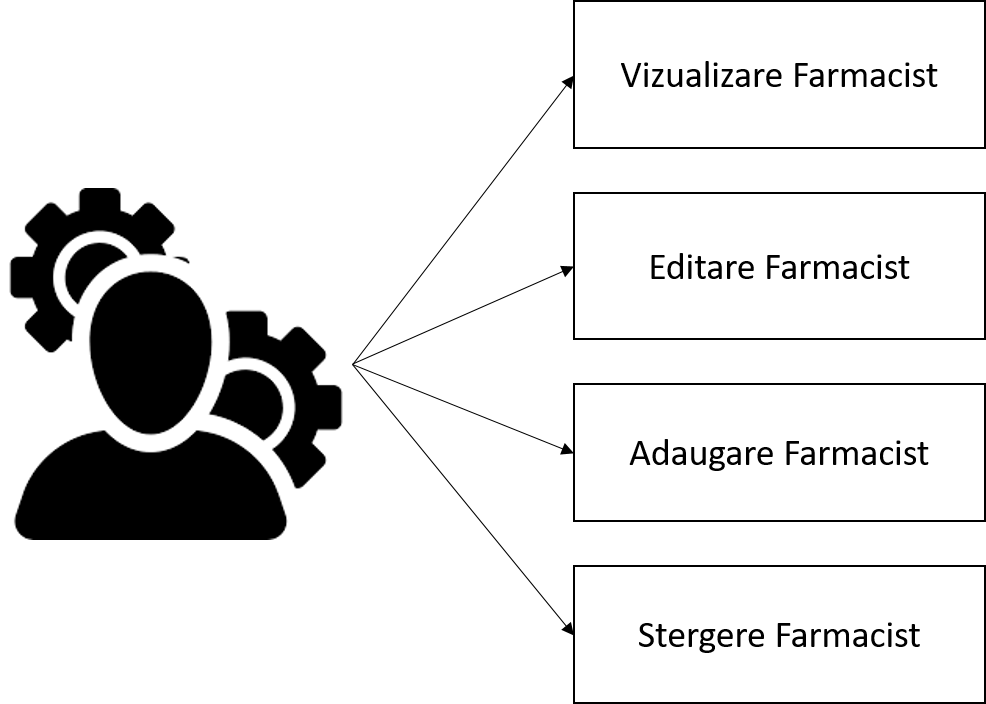
Level: summary level

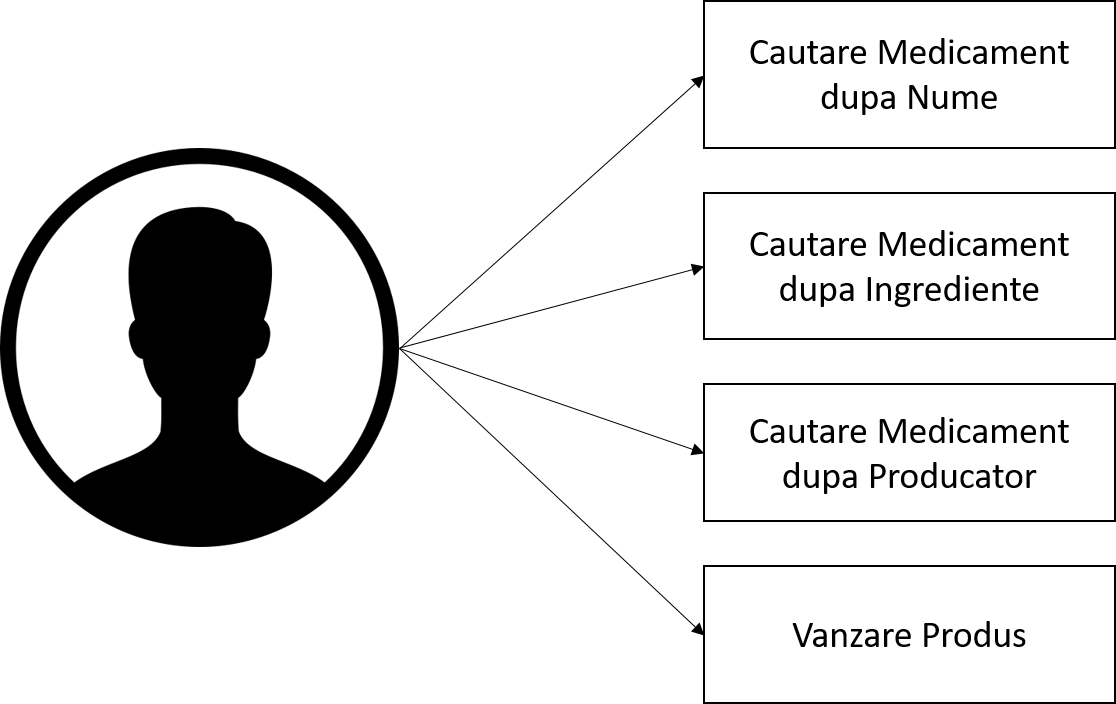
Primary actor: administrator

Use case: Operatii pe medicamente

Level: user-goal level

Primary actor: user





3. System Architectural Design

**3.1 Architectural Pattern Description**

In aceasta aplicatie s-a folosit design pattern-ul MVC(Model-View-Controller). Succesul modelului se datoreaza izolarii logicii de business fata de considerentele interfetei cu utilizatorul, rezultand o aplicatie unde aspectul vizual sau/si nivelele inferioare ale logicii de business sunt mai usor de modificat, fara a afecta alte nivele.

**

*Schema conceptuala a modelului.*

**3.2 Diagrams**

Diagrama de pachete:

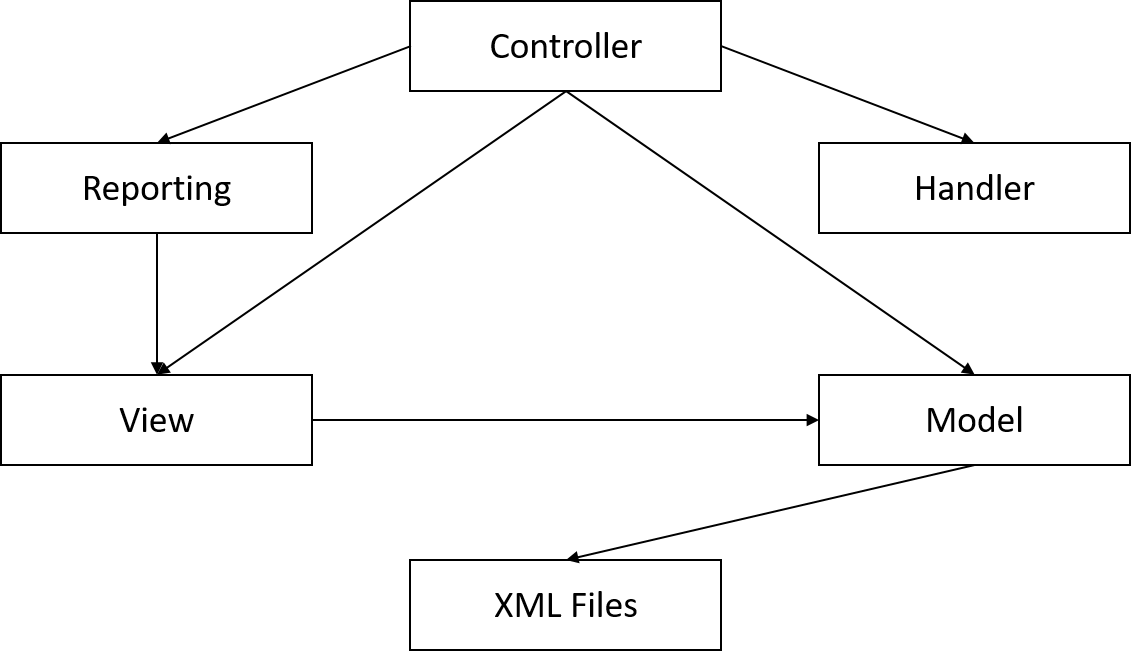


Diagrama de componente pentru AdminController:

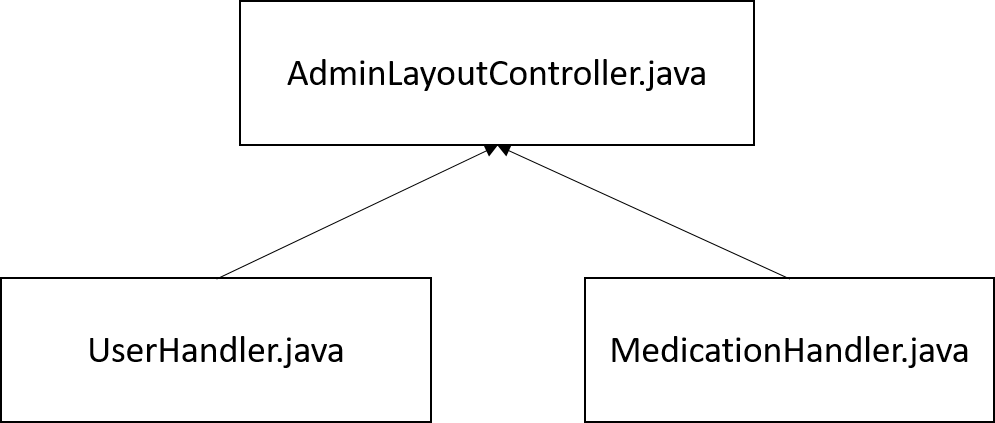
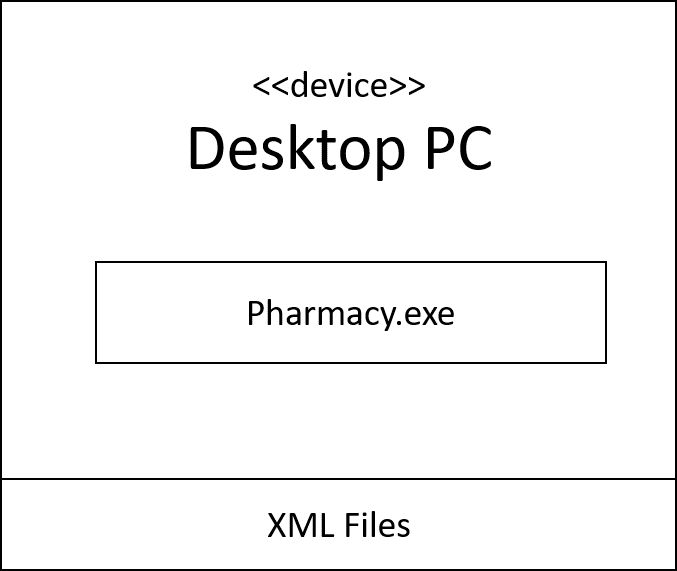
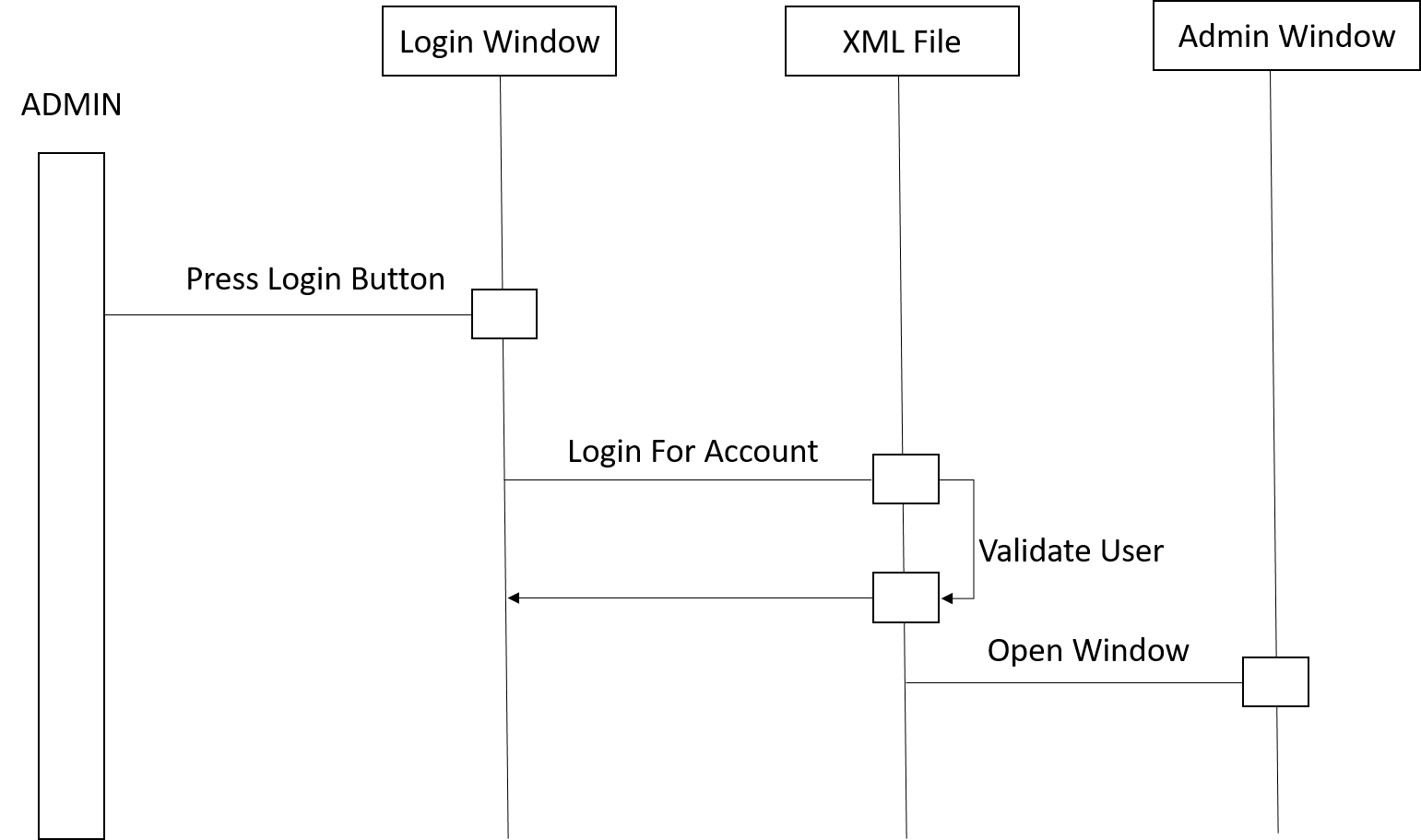


Diagrama Deployment:



4. UML Sequence Diagrams

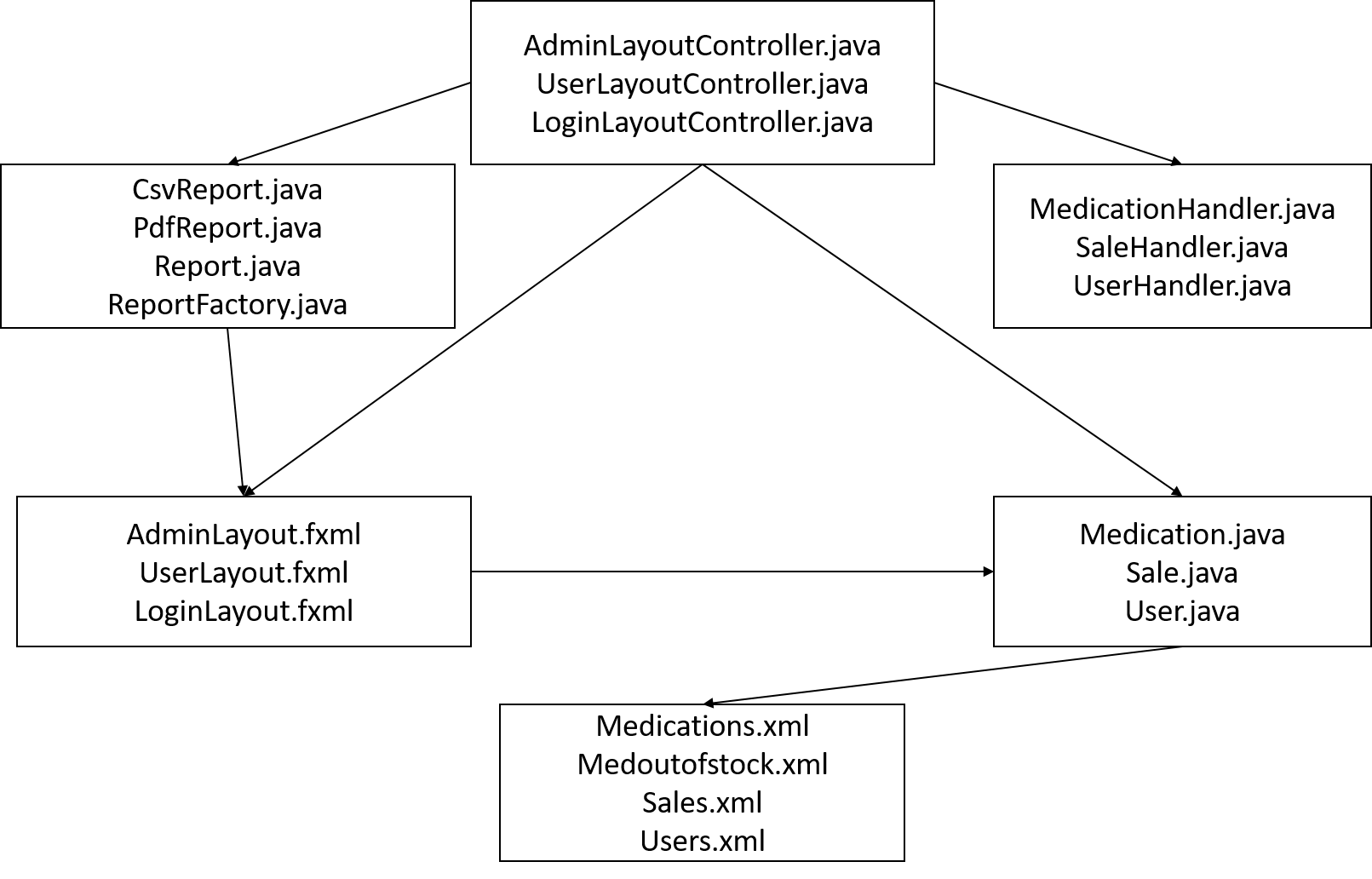


5. Class Design

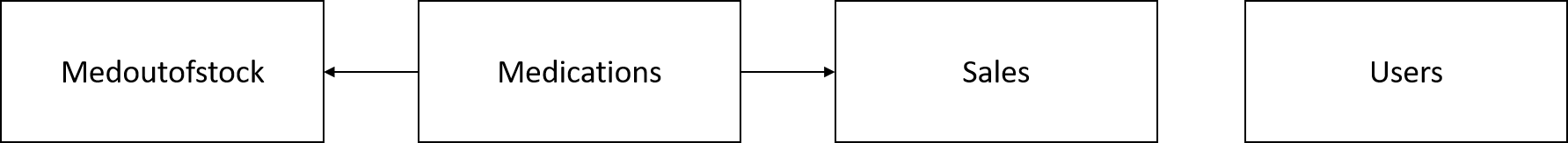
**5.1 Design Patterns Description**

Pe langa *MVC* a mai fost folosit si design pattern-ul *Factory Method* pentru generarea raporturilor in cele doua formate, *pdf* si *csv*. Asadar a fost create o interfata *Report* continand o metoda *generateReport*. Aceasta interfata este implementata de doua clase, *PdfReport* si *CsvReport* ambele avand propria implementare pentru metoda *generateReport*. In final a fost create clasa *ReportFactory* care contine metoda *getReport*. Aceasta metoda returneaza o instanta de *PdfReport* sau de *CsvReport* in functie de valoarea parametrului *reportType* care este de tip *String*.

**5.2 UML Class Diagram**



6. Data Model



7. System Testing

Aceasta aplicatie are numeroase validari. Aceste validari sunt vizibile chiar si din interfata utilizator prin aparitia unor *ferestre de dialog* de atentionare atunci cand datele introduse sunt incorecte. Spre exemplu daca *username-ul* si/sau *parola* unui user nu sunt introduse corect o fereastra de dialog apare pe ecran cu un mesaj adecvat astfel incat utilizatorul sa isi poata da seama ce a gresit. Numeroase validari apar si la vanzarea medicamentelor. De exemplu atunci cand se incearca vanzarea unui medicament in numar mai mare decat este in stoc pe ecran apare o fereastra de dialog care spune ca stocul este insuficient.

8. Bibliography

<http://code.makery.ch/library/javafx-8-tutorial>

<https://docs.oracle.com/javase/8/docs/api>

<https://ro.wikipedia.org/wiki/Model-view-controller>

<https://www.tutorialspoint.com/design_pattern/factory_pattern.htm>

<http://thinktibits.blogspot.ro/2011/05/java-itext-convert-xml-to-pdf-example.html>

<https://stackoverflow.com/questions/21413978/convert-an-xml-file-to-csv-file-using-java>

<https://www.mkyong.com/java/how-to-read-xml-file-in-java-dom-parser/>